

Python: module genutil.selval

genutil.selval

[index](#)

Modules

[MV](#)

[Numeric](#)

[cdtime](#)

Classes

[cdms.selectors.SelectorComponent](#)
[PickComponent](#)

class **PickComponent**([cdms.selectors.SelectorComponent](#))

Let the user pick non contiguous values along an axis
keyword "match" is reserved for handling of inexisting values
match=1 : (default): raise an exception if one of the select-values
match=0 : replace inexistince selctet-values with missing
match=-1: skip inexisting select-values

Methods defined here:

__init__(self, *args, **kargs)

 initialise some value such as tolerances for equality

__str__(self)

post(self, fetched, slab, axes, specifications, confined_by, aux, axismap)

 Post processing retouches the bounds and later will deal with

specify(self, slab, axes, specification, confined_by, aux)

 First part: confine the slab within a Domain wide enough to c

Methods inherited from [cdms.selectors.SelectorComponent](#):

specifyGrid(self, var, grid, specs)

 Refine the specification suitable for grid.intersect().

 'var' is a variable.

 'grid' is the grid associated with the variable.

 'specs' is the result set of specifications, of the form defi

```
Return:  
    0 if self confines the grid.  
    1 if self is not associated with coordinate regions, or does  
  
Note: This function should return 0 only if self is a component of a rectilinear grid. For nonrectilinear grids, see class coordinateComponent.
```

Functions

picker(*args, **kargs)

Let the user pick non contiguous values along an axis

Usage:

```
picker(dim1=list1, dim2=list2)
```

keyword 'match' is reserved for handling of inexisting values

```
match=1 : (default): raise an exception if one of the select-values
```

```
match=0 : replace inexistince selct-values with missing
```

```
match=-1: skip inexisting select-values
```

Example:

```
f=cdms.open('/pcmdi/obs/mo/ta/rnl_ncep/ta.rnl_ncep.ctl')  
#f first levels are 1000.00, 925.00, 850.00, 700.00  
s=f('ta,picker(level=[1000,850,700]))  
#or  
s=f('ta,picker(level=[1000,700,850]) # different order from first  
#or  
s=f('ta,picker(level=[1000,700,800]) # raise an exception since 800  
#or  
s=f('ta,picker(level=[1000,700,800],match=0) # replace 800 level w  
#or  
s=f('ta,picker(level=[1000,700,800],match=-1) # skip 800 level  
# or  
s=f('ta',genutil.picker(time=['1987-7','1988-3'],cdtime.comptime(1987,7,1),cdtime.comptime(1988,3,1)))
```